The Baire partial quasi-metric space: A mathematical tool for asymptotic complexity analysis in Computer Science

M.A. Cerdà-Uguet[1], M.P. Schellekens[2], O. Valero[1*]

[1]Departamento de Ciencias Matemáticas e Informática,
Universidad de las Islas Baleares, 07122, Baleares, Spain
e-mails: macerda2@educacio.caib.es, o.valero@uib.es
[2]Center of Efficiency-Oriented Languages, Department of
Computer Science, University College Cork, Western Road,
Cork, Ireland
e-mail: m.schellekens@cs.ucc.ie

Abstract

In 1994, S.G. Matthews introduced the notion of partial metric space in order to obtain a suitable mathematical tool for program verification [Ann. New York Acad. Sci. 728 (1994), 183-197]. He gave an application of this new structure to parallel computing by means of a partial metric version of the celebrated Banach fixed point theorem [Theoret. Comput. Sci. 151 (1995), 195-205]. Later on, M.P. Schellekens introduced the theory of complexity (quasi-metric) spaces as a part of the development of a topological foundation for the asymptotic complexity analysis of programs and algorithms [Electronic Notes in Theoret. Comput. Sci. 1 (1995), 211-232]. The applicability of this theory to the asymptotic complexity analysis of Divide and Conquer algorithms was also illustrated by Schellekens. In

---

[*]Corresponding author e-mail, telephone and fax numbers: o.valero@uib.es, +34 971 259817, +34 971 173003

1

particular, he gave a new proof, based on the use of the aforenamed Banach fixed point theorem, of the well-known fact that Mergesort algorithm has optimal asymptotic average running time of computing. In this paper, motivated by the utility of partial metrics in Computer Science, we discuss whether the Matthews fixed point theorem is a suitable tool to analyze the asymptotic complexity of algorithms in the spirit of Schellekens. Specifically, we show that a slight modification of the well-known Baire partial metric on the set of all words over an alphabet constitutes an appropriate tool to carry out the asymptotic complexity analysis of algorithms via fixed point methods without the need for assuming the convergence condition inherent to the definition of the complexity space in the Shellekens framework. Finally, in order to illustrate and to validate the developed theory we apply our results to analyze the asymptotic complexity of Quicksort, Mergesort and Largesort algorithms.

2010 AMS Classification: 47H10, 54E50, 54F05, 68Q25, 68W40.

Keywords: asymptotic complexity analysis, recurrence equation, quasi-metric, partial metric, Baire partial metric, Baire partial quasi-metric, running time of computing, fixed point.

1   Introduction and preliminaries

Throughout this paper the letters $\mathbb{R}^+$, $\mathbb{N}$ and ω will denote the set of nonnegative real numbers, the set of positive integer numbers and the set of nonnegative integer numbers, respectively.

In Computer Science the complexity analysis of an algorithm is based on determining mathematically the quantity of resources needed by the algorithm in order to solve the problem for which it has been designed. A typical resource, playing a central role in complexity analysis, is the execution time or running time of computing. Since there are often many algorithms to solve the same problem, one objective of the complexity analysis is to assess which of them is faster when large inputs are considered. To this end, it is necessary to compare their running time of computing. This is usually done by means of the asymptotic analysis in which the running time of an algorithm is denoted by a function $T : \mathbb{N} \to (0, \infty]$ in such a way that $T(n)$ represents the time taken by the algorithm to solve the problem under consideration when the input of the algorithm is of size n. Let us denote by $\mathcal{RT}$ the set

formed by all functions from $\mathbb{N}$ into $(0, \infty]$. Of course the running time of an algorithm does not only depend on the input size n, but it depends also on the particular input of the size n (and the distribution of the data). Thus the running time of an algorithm is different when the algorithm processes certain instances of input data of the same size n. As a consequence, for the purpose of size-based comparisons, it is necessary to distinguish three possible behaviours in the complexity analysis of algorithms. These are the so-called best case, the worst case and the average case. The best case and the worst case for an input of size n are defined by the minimum and the maximum running time of computing over all inputs of the size n, respectively. The average case for an input of size n is defined by the expected value or average over all inputs of the size n.

In general, to determine exactly the function which describes the running time of computing of an algorithm is an arduous task. However, in most situations is sufficient to know the running time of computing of an algorithm in an "approximate" way rather than in an exact one. For this reason the asymptotic complexity analysis of algorithms, based on the $\mathcal{O}$-notation, is focused on obtaining the "approximate" running time of computing of an algorithm. Indeed, if $g \in \mathcal{RT}$ denotes the running time of computing of an algorithm then the statement $g \in \mathcal{O}(f)$, where $f \in \mathcal{RT}$, means that there exists $n_0 \in \mathbb{N}$ and $c \in \mathbb{R}^+$ such that $g(n) \leq cf(n)$ for all $n \in \mathbb{N}$ such that $n \geq n_0$ ($\leq$ stands for the usual order on $\mathbb{R}^+$). So the function f gives an asymptotic upper bound of the running time g and, thus, an "approximate" information of the running time of the algorithm. The set $\mathcal{O}(f)$ is called the asymptotic complexity class of f. Hence, from an asymptotic complexity analysis viewpoint, determining the running time of an algorithm consists of obtaining its asymptotic complexity class. For a fuller treatment of complexity analysis of algorithms we refer the reader to [1, 3].

In 1995, M.P. Schellekens introduced the theory of complexity spaces as a part of the development of a topological foundation for the asymptotic complexity analysis of algorithms ([14]). This theory is based on the notion of quasi-metric space.

Let us recall that, following [8], a quasi-metric on a nonempty set X is a function $d : X \times X \to \mathbb{R}^+$ such that for all $x, y, z \in X$:

$(i) \quad d(x,y) = d(y,x) = 0 \Leftrightarrow x = y;$
$(ii) \quad d(x,y) \leq d(x,z) + d(z,y).$

A quasi-metric space is a pair (X, d) such that X is a nonempty set and

3

d is a quasi-metric on X.

Each quasi-metric d on X generates a T_0-topology $\mathcal{T}(d)$ on X which has as a base the family of open d-balls $\{B_d(x,\varepsilon) : x \in X,\ \varepsilon > 0\}$, where $B_d(x,\varepsilon) = \{y \in X : d(x,y) < \varepsilon\}$ for all $x \in X$ and $\varepsilon > 0$.

Given a quasi-metric d on X, the function d^s defined on $X \times X$ by $d^s(x,y) = \max(d(x,y), d(y,x))$ is a metric on X.

A quasi-metric space (X,d) is called bicomplete if the metric space (X,d^s) is complete.

A well-known example of a bicomplete quasi-metric space is the pair $((0,\infty], u_{-1})$, where

$$u_{-1}(x,y) = \max\left(\frac{1}{y} - \frac{1}{x}, 0\right)$$

for all $x, y \in (0,\infty]$.

The quasi-metric space $((0,\infty], u_{-1})$ plays a central role in the Schellekens theory. Indeed, let us recall that the complexity space is the pair $(\mathcal{C}, d_{\mathcal{C}})$, where

$$\mathcal{C} = \{f \in \mathcal{RT} : \sum_{n=1}^{\infty} 2^{-n} \frac{1}{f(n)} < \infty\}$$

and $d_{\mathcal{C}}$ is the quasi-metric on $\mathcal{C}$ defined by

$$d_{\mathcal{C}}(f,g) = \sum_{n=1}^{\infty} 2^{-n} \max\left(\frac{1}{g(n)} - \frac{1}{f(n)}, 0\right).$$

Obviously we adopt the convention that $\frac{1}{\infty} = 0$.

According to [14], from a complexity analysis point of view, it is possible to associate a function of $\mathcal{C}$ with each algorithm in such a way that such a function represents, as a function of the size of the input data, the running time of computing of the algorithm. Because of this, the elements of $\mathcal{C}$ are called complexity functions. Moreover, given two functions $f, g \in \mathcal{C}$, the numerical value $d_{\mathcal{C}}(f,g)$ (the complexity distance from f to g) can be interpreted as the relative progress made in lowering the complexity by replacing any program P with complexity function f by any program Q with complexity function g. Therefore, if $f \neq g$, the condition $d_{\mathcal{C}}(f,g) = 0$ can be read as f is "at least as efficient" as g on all inputs (i.e. $d_{\mathcal{C}}(f,g) = 0 \Leftrightarrow f(n) \leq g(n)$ for all $n \in \mathbb{N}$). Thus we can encode the natural order relation on the set $\mathcal{C}$,

induced by the pointwise order $\leq$, through the quasi-metric $d_{\mathcal{C}}$. In particular the fact that $d_{\mathcal{C}}(f,g) = 0$ implies that $f \in \mathcal{O}(g)$.

The applicability of this theory to the asymptotic complexity analysis of Divide and Conquer algorithms was illustrated by Schellekens in [14]. In particular, he gave a new proof of the well-known fact that Mergesort algorithm has optimal asymptotic average running time of computing. To do this he introduced a method, based on a fixed point theorem for functionals defined on the complexity space into itself, to analyze the general class of Divide and Conquer algorithms. Let us recall the aforenamed method.

A Divide and Conquer algorithm solves a problem of size n ($n \in \mathbb{N}$) splitting it into a subproblems of size $\frac{n}{b}$, for some constants a, b with $a, b \in \mathbb{N}$ and $a, b > 1$, and solving them separately by the same algorithm. After obtaining the solution of the subproblems, the algorithm combines all subproblem solutions to give a global solution to the original problem. The recursive structure of a Divide and Conquer algorithm leads to a recurrence equation for the running time of computing. In many cases the running time of a Divide and Conquer algorithm is the solution to a recurrence equation of the form

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ aT(\frac{n}{b}) + h(n) & \text{if } n \in \omega_b \end{cases}, \qquad (1)$$

where $\omega_b = \{b^k : k \in \mathbb{N}\}$, $c \in \mathbb{R}^+$ ($c > 0$) denotes the complexity on the base case (i.e. the problem size is small enough and the solution takes constant time), $h(n)$ represents the time taken by the algorithm in order to divide the original problem into a subproblems and to combine all subproblems solutions into a unique one ($h \in \mathcal{RT}$ and $0 < h(n) < \infty$ for all $n \in \mathbb{N}$).

Notice that for Divide and Conquer algorithms, it is typically sufficient to obtain the complexity on inputs of size n where n ranges over the set ω_b ([1, 3, 14]).

Typical examples of algorithms whose running time of computing can be obtained by means of the recurrence (1) are Quicksort (in the best case behaviour) and Mergesort.

In order to compute the running time of computing of a Divide and Conquer algorithm satisfying the recurrence equation (1), it is necessary to show that such a recurrence equation has a unique solution and, later, to obtain the asymptotic complexity class of such a solution. The method introduced by Schellekens to show that the equation (1) has a unique solution, and to obtain the asymptotic complexity class of the solution is outlined

below:

Denote by $\mathcal{C}_{b,c}$ the subset of $\mathcal{C}$ given by

$$\mathcal{C}_{b,c} = \{f \in \mathcal{C} : f(1) = c \text{ and } f(n) = \infty \text{ for all } n \notin \omega_b \text{ with } n > 1\}.$$

Since the quasi-metric space $(\mathcal{C}, d_{\mathcal{C}})$ is bicomplete (see Theorem 3 and Remark in page 317 of [13]) and the set $\mathcal{C}_{b,c}$ is closed in $(\mathcal{C}, d_{\mathcal{C}}^s)$, we have that the quasi-metric space $(\mathcal{C}_{b,c}, d_{\mathcal{C}}|_{\mathcal{C}_{b,c}})$ is bicomplete.

Next we associate a functional $\Phi_T : \mathcal{C}_{b,c} \to \mathcal{C}_{b,c}$ with the recurrence equation (1) of a Divide and Conquer algorithm defined as follows:

$$\Phi_T(f)(n) = \begin{cases} c & \text{if } n = 1 \\ \infty & \text{if } n \notin \omega_b \text{ and } n > 1 \\ af(\frac{n}{b}) + h(n) & \text{otherwise} \end{cases} . \tag{2}$$

Of course a complexity function in $\mathcal{C}_{b,c}$ is a solution to the recurrence equation (1) if and only if it is a fixed point of the functional Φ_T. It was proved in [14] that

$$d_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(\Phi_T(f), \Phi_T(g)) \leq \frac{1}{a} d_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(f,g) \tag{3}$$

for all $f, g \in \mathcal{C}_{b,c}$. So by Banach's fixed point theorem for metric spaces we deduce that the functional $\Phi_T : \mathcal{C}_{b,c} \to \mathcal{C}_{b,c}$ has a unique fixed point and, thus, the recurrence equation (1) has a unique solution.

In order to obtain the asymptotic complexity class of the solution to the recurrence equation (1), Schellekens introduced a special class of functionals known as improvers.

Let $C \subseteq \mathcal{RT}$, a functional $\Phi : C \to C$ is called an *improver* with respect to a function $f \in C$ provided that $\Phi^{n+1}(f) \leq \Phi^n(f)$ for all $n \in \omega$. Of course $\Phi^0(f) = f$.

Observe that an improver is a functional which corresponds to a transformation on programs in such a way that the iterative applications of the transformation yield an improved, from a complexity point of view, program at each step of the iteration.

Note that when Φ is monotone, to show that Φ is an improver with respect to $f \in C$, it suffices to verify that $\Phi(f) \leq f$.

Under these conditions the following result was stated in [14]:

Theorem 1. *A Divide and Conquer recurrence of the form (1) has a unique solution f_T in $\mathcal{C}_{b,c}$. Moreover, if the monotone functional Φ_T associated to (1), and given by (2), is an improver with respect to some function $g \in \mathcal{C}_{b,c}$, then the solution to the recurrence equation satisfies that $f_T \in \mathcal{O}(g)$.*

In [14] Schellekens discussed the complexity class of Mergesort in order to illustrate the utility of Theorem 1. In the particular case of Mergesort the recurrence equation (1) in the average case behaviour is exactly

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\frac{n}{2}) + \frac{n}{2} & \text{if } n \in \omega_2 \end{cases}. \qquad (4)$$

Of course Theorem 1 provides that the recurrence equation (4) has a unique solution $f_T \in \mathcal{C}_{2,c}$. In addition, Schellekens proved in [14] that the functional Φ_T induced by the recurrence equation (4) is an improver with respect to a complexity function $g_k \in \mathcal{C}_{b,c}$, with $k \in \mathbb{R}^+$ and $g_k(n) = kn \log_2(n)$ for all $n \in \omega_2$, if and only if $\frac{1}{2} \leq k$. Therefore, by Theorem 1, we conclude that $f \in \mathcal{O}(g_{\frac{1}{2}})$, i.e. Theorem 1 provides a formal proof of the well-known fact that the running time of computing of Mergesort in the average case behaviour is in the asymptotic complexity class of $n \log_2(n)$.

When a program uses a recursion process to find the solution to a problem, such a process is characterized by obtaining in each step of the computation an approximation to the aforementioned solution which is better than the approximations obtained in the preceding steps and, in addition, by obtaining always the final approximation to the problem solution as the "limit" of the computing process. A mathematical model to this sort of situations was developed by D.S. Scott which is based on ideas from order theory and topology (see, for instance, [15], [16] and [5]). In particular, the order represents some notion of information in such a way that each step of the computation is identified with an element of the mathematical model which is greater than (or equal to) the other ones associated with the preceding steps, since each approximation gives more information about the final solution than those computed before. The final output of the computational process is seen as the limit of the successive approximations. Thus the recursion processes are modeled as increasing sequences of elements of the ordered set which converge to its least upper bound with respect to the given topology. From an Information Theory point of view the Scott model needs to distinguish two kind of elements: the so-called totally defined objects and the partially defined objects. The former represent elements of the model whose information content can not be stored in a computer and, therefore, must be approximated. The partially defined objects match with those elements of the model that can be stored in the computer and that approximate the total ones.

In 1994 S.G. Matthews introduce the notion of partial metric as a mathematical tool to model computational processes in the spirit of Scott where

a quantitative degree of the information content of the involved elements is needed.

Let us recall some notions related to partial metrics in order to explain the importance of this new metric concept.

According to [9], a partial metric on a nonempty set X is a function $p : X \times X \to \mathbb{R}^+$ such that for all $x, y, z \in X$:

(i) $p(x, x) = p(x, y) = p(y, y) \Leftrightarrow x = y$;
(ii) $p(x, x) \leq p(x, y)$;
(iii) $p(x, y) = p(y, x)$;
(iv) $p(x, y) \leq p(x, z) + p(z, y) - p(z, z)$.

A partial metric space is a pair (X, p) such that X is a nonempty set and p is a partial metric on X.

Each partial metric p on X generates a T_0 topology $\mathcal{T}(p)$ on X which has as a base the family of open p-balls $\{B_p(x, \varepsilon) : x \in X, \varepsilon > 0\}$, where $B_p(x, \varepsilon) = \{y \in X : p(x, y) < p(x, x) + \varepsilon\}$ for all $x \in X$ and $\varepsilon > 0$. From this fact it immediately follows that a sequence $(x_n)_{n \in \mathbb{N}}$ in a partial metric space (X, p) converges to a point $x \in X$ with respect to $\mathcal{T}(p) \Leftrightarrow p(x, x) = \lim_{n \to \infty} p(x, x_n)$.

As usual ([5]) an order on a (nonempty) set X is a reflexive, antisymmetric and transitive binary relation $\leq$ on X, and a set equipped with an order is said to be an ordered set.

The success of partial metrics lies in that every partial metric p induces an order $\leq_p$ on X ($x \leq_p y \Leftrightarrow p(x, y) = p(x, x)$) in such a way that increasing sequences of elements with respect to $\leq_p$ converge to its least upper bound with respect the partial metric topology $\mathcal{T}(p)$. Moreover, partial metrics can be used to distinguish between objects with totally defined information content and objects with partially defined information content in Scott's models. Specifically if (X, p) is a partial metric space, then the numerical value $p(x, x)$ allows us to quantify the amount of information contained in an object x. In particular the smaller $p(x, x)$ the more defined x is, being x totally defined if $p(x, x) = 0$.

Motivated by the usefulness of partial metric spaces in Computer Science, it seems natural to wonder if these kind of metric spaces are also useful in complexity analysis in the spirit of Schellekens. However we show in the following that the preceding question has at first a negative answer. To this end, let us recall some additional and useful concepts about partial metrics.

Following [9], a sequence $(x_n)_{n \in \mathbb{N}}$ in a partial metric space (X, p) is called

a Cauchy sequence if $\lim_{n,m \to \infty} p(x_n, x_m)$ exists. A partial metric space (X, p) is said to be complete if every Cauchy sequence $(x_n)_{n \in \mathbb{N}}$ in X converges, with respect to $\mathcal{T}(p)$, to a point $x \in X$ such that $p(x, x) = \lim_{n,m \to \infty} p(x_n, x_m)$.

Inspired, in part, by the applications to program verification Matthews extends Banach's fixed point theorem to the framework of partial metric spaces, and he used it to formulate a suitable test for lazy data flow deadlock in Khan's model of parallel computation ([10]). The aforementioned new theorem can be stated as follows:

Theorem 2. *Let f be a mapping of a complete partial metric space (X, p) into itself such that there is $s \in \mathbb{R}^+$ with $0 \le s < 1$, satisfying*

$$p(f(x), f(y)) \le s p(x, y),$$

for all $x, y \in X$. Then f has a unique fixed point. Moreover if $x \in X$ is the fixed point of f, then $p(x, x) = 0$.

According to [9] and [11] some correspondences between quasi-metric and partial metric spaces can be stated.

Proposition 3. *If (X, p) is a partial metric space, then the following statements hold:*

(i) The function $d_p : X \times X \to \mathbb{R}^+$ defined by $d_p(x, y) = p(x, y) - p(x, x)$ is a quasi-metric on X such that $\mathcal{T}(p) = \mathcal{T}(d_p)$.

(ii) The below assertions are equivalent:

(1) (X, p) is complete

(2) (X, d_p) is bicomplete.

Following [11] the set $\mathcal{C}$ can be endowed with a partial metric $p_{\mathcal{C}}$ defined for all $f, g \in \mathcal{C}$ by

$$p_{\mathcal{C}}(f, g) = \sum_{n=1}^{\infty} 2^{-n} \max \left(\frac{1}{f(n)}, \frac{1}{g(n)} \right).$$

Moreover it is not hard to see that the partial metric $p_{\mathcal{C}}$ induces, by Proposition 3, the quasi-metric $d_{\mathcal{C}}$. So Proposition 3 guarantees that the partial metric space $(\mathcal{C}, p_{\mathcal{C}})$ is complete. Also $(\mathcal{C}_{b,c}, p_{\mathcal{C}}|_{\mathcal{C}_{b,c}})$ is complete, since $(\mathcal{C}_{b,c}, d_{\mathcal{C}}|_{\mathcal{C}_{b,c}})$ is bicomplete.

9

Now suppose that there exists $s \in \mathbb{R}^+$ with $0 \leq s < 1$ such that

$$p_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(\Phi_T(f), \Phi_T(g)) \leq s p_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(f, g).$$

for all $f, g \in \mathcal{C}_{b,c}$.

We only consider the case of $0 < s < 1$, because it is evident that the case $s = 0$ gives a contradiction.

Take $f, g \in \mathcal{C}_{b,c}$ defined by $f(n) = 2c$ and $g(n) = 2(c + 1)$ for all $n \in \omega_b$. It is clear that

$$p_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(\Phi_T(f), \Phi_T(g)) = \frac{1}{2c} + \sum_{n=2}^{\infty} 2^{-b^n} \max\left(\frac{1}{2ac + h(b^n)}, \frac{1}{2a(c+1) + h(b^n)}\right).$$

Moreover,

$$
\begin{aligned}
p_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(f, g) &= \sum_{n=1}^{\infty} 2^{-b^n} \max\left(\frac{1}{f(b^n)}, \frac{1}{g(b^n)}\right) \leq \sum_{n=1}^{\infty} 2^{-n} \max\left(\frac{1}{2c}, \frac{1}{2(c+1)}\right) \\
&= \frac{1}{2c}.
\end{aligned}
$$

Applying the hypothesis we obtain that

$$\frac{1}{2c} \leq p_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(\Phi_T(f), \Phi_T(g)) \leq s p_{\mathcal{C}}|_{\mathcal{C}_{b,c}}(f, g) \leq s\frac{1}{2c}.$$

As a result we deduce that $1 \leq s < 1$, which is a contradiction.

Consequently Theorem 2 can not be used to analyze the complexity of the algorithms whose running time of computing is associated to a recurrence equation (1) when the partial metric $p_{\mathcal{C}}$, instead the quasi-metric $d_{\mathcal{C}}$, is employed as a complexity distance.

In the light of the preceding conclusion our purpose in this paper is to demonstrate that partial metric spaces, and in particular the partial metric fixed point theorem, can be used satisfactorily for the asymptotic complexity analysis of algorithms in the spirit of Schellekens' approach based on improvers. To achieve this goal we focus our attention on a slight modification, that we have called Baire partial quasi-metric, of the well-known Baire partial metric on the set of all words over an alphabet. We show that such a partial metric tool constitutes an appropriate implement to carry out formally the asymptotic complexity analysis of algorithms whose running time of computing leads to various types of recurrences equations (not only the Divide and

10

Conquer ones). In particular, we validate the usefulness of our new results by means of applying them to analyze the well-known asymptotic complexity of several illustrative algorithms such as Quicksort, Mergesort and Largetwo.

The remainder of the paper is organized as follows: Section 2 is devoted to introduce the noted Baire partial metric and, in addition, to prove that it allows to show the existence of solution to recurrence equations that arise in a natural way in complexity analysis of algorithms. However, in the same section we show that the Baire partial metric can not be employed, in general, to obtain an asymptotic upper bound, and thus the complexity class, of the running time of computing of an algorithm. Inspired by the last fact, we introduce in Section 3 a new Baire partial metric framework whose basis resides in the partial quasi-metric approach introduced recently in [7]. We show that this new partial metric approach presents a relevant advantage with respect to the Schellekens one. More specifically, it is suitable to carry out the asymptotic complexity analysis of algorithms via fixed point methods without the need for assuming the convergence condition inherent to the definition of the complexity space in the Shellekens framework. Finally, the discussion of the running time of the celebrated Quicksort, Mergesort and Largetwo allows us, on one hand, to validate our new results and, on the other hand, to show the potential applicability of the developed theory to complexity analysis in Computer Science.

2 The Baire partial metric space and the asymptotic complexity analysis of algorithms

With the aim of applying partial metric spaces to complexity analysis of algorithms we recall the well-known Baire partial metric space which was used by Matthews to model Kahn's parallel computation processes ([6, 10]).

Let Σ be a nonempty set (an alphabet). Denote by Σ^∞ the set of all finite and infinite sequences (words) over Σ.

For each $x \in \Sigma^\infty$ we denote by $l(x)$ the length of x. Hence $l(x) \in [1, \infty]$.

From now on, if $x \in \Sigma^\infty$ with $l(x) = \infty$ we will write $x := x_1 x_2 \ldots$, and if $x \in \Sigma^\infty$ with $l(x) = n < \infty$ we will write $x := x_1 x_2 \ldots x_n$.

Given $x, y \in \Sigma^\infty$, denote by $l(x, y)$ the length of the longest common prefix of x and y, i.e. $l(x, y) = \sup\{n \in \mathbb{N} : x_k = y_k \text{ whenever } k \leq n\}$ if x and y have a common prefix, and $l(x, y) = 0$ otherwise.

As usual the set Σ^∞ is equipped with the prefix order $\sqsubseteq$, which is defined by $x \sqsubseteq y \Leftrightarrow x$ is a prefix of y.

Define on $\Sigma^\infty \times \Sigma^\infty$ the nonnegative real valued function p_B by

$$p_B(x, y) = 2^{-l(x,y)}$$

for all $x, y \in \Sigma^\infty$. Of course it is adopted the convention that $2^{-\infty} = 0$. It is not hard to see that the pair (Σ^∞, p_B) is a complete partial metric space ([11]), which is known as the Baire partial metric space.

The partial metric p_B can be used to distinguish between infinite words (objects with totally defined information content) and finite words (objects with partially defined information content) because $p_B(x, x) = 0$ for some $x \in \Sigma^\infty \Leftrightarrow l(x) = \infty$. The fact that the value $p_B(x, x)$ allows to assign a degree of information content plays a crucial role in Information Theory and, in addition, presents an advantage with respect to the role played by the classical metrics extensively used before, as for instance the well-known Baire metric (for a fuller treatment of the classical Baire metric we refer the reader to [2, 12]).

The Baire partial metric space (Σ^∞, p_B) induces, by Proposition 3, the quasi-metric $d_{p_B} : \Sigma^\infty \times \Sigma^\infty \to \mathbb{R}^+$ given by

$$d_{p_B}(x, y) = 2^{-l(x,y)} - 2^{-l(x)}$$

for all $x, y \in \Sigma^\infty$. Note that, similarly to the case of complexity space (see Section 1), we have $d_{p_B}(x, y) = 0 \Leftrightarrow x \sqsubseteq y$.

The next result provides that the Baire partial metric space is suitable to show that the recurrence (1) has a unique solution.

Theorem 4. *Let $\Sigma = (0, \infty]$ and $a, b \in \mathbb{N}$ with $a, b > 1$. Fix $z \in \Sigma^\infty$ with $l(z) = \infty$ and $z_k \neq \infty$ for all $k \in \omega_b$ and $k \geq 2$. Let $\Sigma_{b,c}^\infty$ be the subset of Σ^∞ given by $\Sigma_{b,c}^\infty := \{y \in \Sigma^\infty : 2 \leq l(y)$ and $y_1 = c, y_k = \infty$ for all $k \notin \omega_b$ with $2 \leq k \leq l(y)\}$. Then the mapping $\Theta_{a,b}^z : \Sigma_{b,c}^\infty \to \Sigma_{b,c}^\infty$ defined by $\Theta_{a,b}^z(x) = x_{\Theta_{a,b}^z}$, where*

$$(x_{\Theta_{a,b}^z})_k := \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_b \text{ and } 2 \leq k \leq l(x)+1 \\ a \cdot x_{\frac{k}{b}} + z_k & \text{if } k \in \omega_b \text{ and } \frac{k}{b} \leq l(x) \end{cases}$$

has a unique fixed point $v \in \Sigma_{b,c}^\infty$ with $l(v) = \infty$.

Proof. It is easy to check that the set $\Sigma_{b,c}^\infty$ is closed in $(\Sigma^\infty, d_{p_B}^s)$. Hence $(\Sigma_{b,c}^\infty, d_{p_B}^s|_{\Sigma_{b,c}^\infty})$ is complete. By Proposition 3 the partial metric space $(\Sigma_{b,c}^\infty, p_B|_{\Sigma_b^\infty})$ is complete.

A straightforward computation shows that the mapping $\Theta_{a,b}^z$ satisfies the inequality

$$p_B|_{\Sigma_b^\infty}(x_{\Theta_{a,b}^z}, y_{\Theta_{a,b}^z}) \leq \frac{1}{2} p_B|_{\Sigma_b^\infty}(x, y)$$

for all $x, y \in \Sigma_{b,c}^\infty$. Then, by Theorem 2, we deduce that the mapping $\Theta_{a,b}^z$ has unique fixed point $v \in \Sigma_{b,c}^\infty$ with $p_B|_{\Sigma_{b,c}^\infty}(v, v) = 0$. Hence $l(x) = \infty$. ∎

Following the ideas introduced in Section 1 let us denote by $\mathcal{RT}_{b,c}$ the set

$$\mathcal{RT}_{b,c} = \{f \in \mathcal{RT} : f(1) = c \text{ and } f(n) = \infty \text{ for all } n \notin \omega_b \text{ with } n > 1\}.$$

Of course if we take $z \in \Sigma^\infty$ with $z_k = h(b^k)$ for all $k \in \mathbb{N}$ in Theorem 4, where h is given as in the recurrence equation (1), then the function $f_v \in \mathcal{RT}_{b,c}$ defined by $f_v(n) = v_n$ for all $n \in \mathbb{N}$, where v is provided by Theorem 4, can be identified with the solution to the Divide and Conquer recurrence equation (1) and, thus, with the running time of computing of Divide and Conquer algorithms. However, to make a complete asymptotic complexity analysis we must give the complexity class of the running time of computing, i.e. the complexity class of f_v. Unfortunately the presented framework is not able to allow us to get this objective. The reason is given by the fact that two words $x, y \in \Sigma^\infty$ with $l(x) = l(y) = \infty$ satisfy $x \sqsubseteq y \Leftrightarrow x = y$. So, according to the above technique, if f, g represent the running time of computing of two algorithms ($f, g \in \mathcal{RT}$), and we identify in a natural way those functions with the words $x^f, x^g \in \Sigma^\infty$ ($\Sigma = (0, \infty]$) such that

$$x_n^f = f(n) \text{ and } x_n^g = g(n) \text{ for all } n \in \mathbb{N},$$

then we have that

$$x^f \sqsubseteq x^g \Leftrightarrow f(n) = g(n) \text{ for all } n \in \mathbb{N}.$$

Therefore we can not obtain an asymptotic upper bound of the running time of computing of an algorithm under analysis by means of the usual prefix order defined on Σ^∞. Note that in spite of the preceding disadvantage, the Baire partial metric framework provides the basis to develop a mathematical formalism for the complexity analysis of algorithms which does not rely on

13

the technical "convergence" assumption incorporated into the definition of the complexity space $\mathcal{C}$, that is

$$f \in \mathcal{C} \Leftrightarrow f \in \mathcal{RT} \text{ and } \sum_{n=1}^{+\infty} 2^{-n} \frac{1}{f(n)} < +\infty.$$

Of course, although every reasonable algorithm must hold the preceding convergence condition, this one is a little artificial and has the unique purpose of guaranteeing the finiteness of the value $d_C(f, g)$.

In the next subsection we propose, as a result of the preceding conclusion, a slight modification of the Baire partial metric which will allow to develop a suitable mathematical framework for asymptotic complexity analysis free from the convergence assumption.

3 The Baire partial quasi-metric space and the asymptotic complexity analysis of algorithms

In the remainder of the paper we introduce a new metric tool on the set Σ^∞ of all words over an alphabet Σ in such a way that a slight modification of Theorem 2 allows us to model satisfactorily the asymptotic complexity analysis of algorithms via fixed point techniques in the spirit of Schellekens.

3.1 The Baire partial quasi-metric

For our proposal we recall a few pertinent concepts.

In [7] H.A. Künzi, H.A Pajooshesh and Schellekens have introduced and studied the notion of partial quasi-metric. Roughly speaking a partial quasi-metric is a partial metric which does not satisfy the symmetry property. More specifically, a partial quasi-metric on a nonempty set X is a function $q : X \times X \to \mathbb{R}^+$ such that for all $x, y, z \in X$:

(i) $q(x, x) \leq q(x, y)$;
(ii) $q(x, x) \leq q(y, x)$;
(iii) $q(x, y) \leq q(x, z) + q(z, y) - q(z, z)$;
(iv) $q(x, x) = q(x, y)$ and $q(y, y) = q(y, x) \Leftrightarrow x = y$.

14

Observe that a partial metric on a set X is a partial quasi-metric satisfying in addition the condition:

$$(v) \quad q(x, y) = q(y, x)$$

for all $x, y \in X$.

A partial quasi-metric space is a pair (X, q) such that X is a nonempty set and q is a partial quasi-metric on X.

Similarly to the case of partial metric spaces a partial quasi-metric q generates a T_0-topology $\mathcal{T}(q)$ on X which has as a base the family of open q-balls $\{B_q(x, \varepsilon) : x \in X, \varepsilon > 0\}$, where $B_q(x, \varepsilon) = \{y \in X : q(x, y) < q(x, x) + \varepsilon\}$ for all $x \in X$ and $\varepsilon > 0$.

If q is a partial quasi-metric on X, then the function $d_q : X \times X \to \mathbb{R}^+$, defined by

$$d_q(x, y) = q(x, y) - q(x, x)$$

for all $x, y \in X$, is a quasi-metric.

On account of [7], a partial quasi-metric space (X, q) is said to be complete provided that the associated quasi-metric space (X, d_q) is bicomplete. Moreover, in the same reference the Matthews fixed point theorem (Theorem 2 in Section 1) has been extended to the context of partial quasi-metric spaces in the following way:

Theorem 5. *Let f be a mapping from a complete partial quasi-metric space (X, q) into itself such that there is $s \in \mathbb{R}^+$ with $0 \leq s < 1$, satisfying*

$$q(f(x), f(y)) \leq sq(x, y), \tag{5}$$

for all $x, y \in X$. Then f has a unique fixed point. Moreover if $x \in X$ is the fixed point of f, then $q(x, x) = 0$.

The below result will be crucial in order to construct a partial quasi-metric framework for the asymptotic complexity analysis.

Proposition 6. *Under the conditions of Theorem 5, if there exists $y \in X$ such that $d_q(f(y), y) = 0$ then $d_q(x, y) = 0$.*

Proof. Suppose for the purpose of contradiction that $d_q(x, y) \neq 0$, where x is the fixed point of f. Then $q(x, y) > 0$. Consequently the inequality (5)

yields

$$
\begin{aligned}
q(x,y) &\leq q(x,f(x)) + q(f(x),y) - q(f(x),f(x)) \\
&= q(f(x),y) \\
&\leq q(f(x),f(y)) + q(f(y),y) - q(f(y),f(y)) \\
&= q(f(x),f(y)) + d_q(f(y),y) \\
&= q(f(x),f(y)) \leq sq(x,y).
\end{aligned}
$$

It follows that $1 \leq s < 1$, which is a contradiction. $\blacksquare$

Now we are able to construct a new metric tool on the set Σ^∞. Indeed, let Σ be a nonempty alphabet endowed with an order $\preceq$. Then, given $x, y \in \Sigma^\infty$, we will say that x is a subprefix of y, denoted by $x \sqsubseteq_{sp} y$, provided that there exists $n_0 \in \mathbb{N}$ with $n_0 \leq l(x)$ such that $x_k \preceq y_k$ for all $k \leq n_0$. Obviously if $x \sqsubseteq y$, then $x \sqsubseteq_{sp} y$.

Note that, contrary to the case of the prefix, the subprefix notion does not induce an order relation on Σ^∞.

Let us denote by $l_\preceq(x,y) = \sup\{n \in \mathbb{N} : x_k \preceq y_k \text{ for all } k \leq n\}$ whenever $x \sqsubseteq_{sp} y$, and $l_\preceq(x,y) = 0$ otherwise. Note that

$$
l_\preceq(x,y) = \infty \Leftrightarrow l(x) = l(y) = \infty \text{ and } x_k \preceq y_k \text{ for all } k \in \mathbb{N}.
$$

Clearly $l_\preceq(x,y) = l(x,y)$ whenever $x \sqsubseteq y$, and $l_\preceq(x,x) = l(x)$ for all $x \in \Sigma^\infty$.

In the light of the preceding definitions we have the following result.

Proposition 7. *Let Σ be an alphabet endowed with an order $\preceq$. Then the pair (Σ^∞, q_B) is a complete partial quasi-metric space, where $q_B : \Sigma^\infty \times \Sigma^\infty \to \mathbb{R}^+$ is defined by $q_B(x,y) = 2^{-l_\preceq(x,y)}$ for all $x, y \in \Sigma^\infty$.*

Proof. Since $l_\preceq(x,y) \leq l(x)$ and $l_\preceq(y,x) \leq l(x)$ for all $x, y \in \Sigma^\infty$ we deduce immediately that $q_B(x,x) \leq q_B(x,y)$ and that $q_B(x,x) \leq q_B(y,x)$ for all $x, y \in \Sigma^\infty$.

Next consider $x, y \in \Sigma^\infty$ such that $q_B(x,x) = q_B(x,y)$ and $q_B(y,y) = q_B(y,x)$. Then $l(x) = l_\preceq(x,y)$ and $l(y) = l_\preceq(y,x)$. It follows that $l(x) = l(y)$ and that $x = y$.

Now we show that

$$
q_B(x,y) \leq q_B(x,z) + q_B(z,y) - q_B(z,z)
$$

for all $x, y, z \in \Sigma^\infty$.

We assume that $l_{\preceq}(x, z) > 0$ and $l_{\preceq}(z, y) > 0$, because otherwise it is clear that the preceding inequality holds. It follows that $l_{\preceq}(x, y) > 0$. Then it suffices to consider that $l_{\preceq}(x, y) \leq \min\left(l_{\preceq}(x, z), l_{\preceq}(z, y)\right)$. But this clearly forces that $l_{\preceq}(x, y) = \min\left(l_{\preceq}(x, z), l_{\preceq}(z, y)\right)$. Consequently

$$
\begin{aligned}
q_B(x, y) &= 2^{-\min\left(l_{\preceq}(x,z), l_{\preceq}(z,y)\right)} \leq 2^{-l_{\preceq}(x,z)} + 2^{-l_{\preceq}(z,y)} - q_B(z, z) \\
&= q_B(x, z) + q_B(z, y) - q_B(z, z).
\end{aligned}
$$

Therefore we have shown that q_B is a partial quasi-metric on Σ^{∞}.

Finally we show that the partial quasi-metric space (Σ^{∞}, q_B) is complete. Indeed, let $(x_n)_{n\in\mathbb{N}}$ be a Cauchy sequence in $(\Sigma^{\infty}, d^s_{q_B})$. Thus, given $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that

$$
\max\left(2^{-l_{\preceq}(x_n, x_m)} - 2^{-l(x_n)}, 2^{-l_{\preceq}(x_m, x_n)} - 2^{-l(x_m)}\right) < \varepsilon
$$

for all $n, m \geq n_0$.

Since $(x_n)_k = (x_m)_k$ for all $k \leq \min\left(l_{\preceq}(x_n, x_m), l_{\preceq}(x_m, x_n)\right)$ we have that

$$
p_B(x_n, x_m) = 2^{-\min\left(l_{\preceq}(x_n, x_m), l_{\preceq}(x_m, x_n)\right)}.
$$

It follows that

$$
d^s_{p_B}(x_n, x_m) \leq 2 \cdot 2^{-\min\left(l_{\preceq}(x_n, x_m), l_{\preceq}(x_m, x_n)\right)} - 2^{-l(x_n)} - 2^{-l(x_m)} < 3\varepsilon
$$

for all $m, n \geq n_0$. Whence $\lim_{n,m\to\infty} d^s_{p_B}(x_n, x_m) = 0$. Thus $(x_n)_{n\in\mathbb{N}}$ is a Cauchy sequence in $(\Sigma^{\infty}, d^s_{p_B})$. Since (Σ^{∞}, p_B) is a complete partial metric space (see Section 2) we have that there exists $x \in \Sigma^{\infty}$ satisfying that

$$
\lim_{n,m\to\infty} p_B(x_n, x_m) = p_B(x, x) = \lim_{n\to\infty} p_B(x, x_n).
$$

Whence we obtain that $\lim_{n\to\infty} d_{p_B}(x, x_n) = \lim_{n\to\infty} d_{p_B}(x_n, x) = 0$. Thus

$$
\lim_{n\to\infty} 2^{-l_{\preceq}(x, x_n)} = 2^{-l(x)},
$$

since

$$
0 \leq 2^{-l_{\preceq}(x, x_n)} - 2^{-l(x)} \leq 2^{-l(x, x_n)} - 2^{-l(x)} < \varepsilon
$$

for all $n \in \mathbb{N}$ such that $n \geq n_0$.

Similar considerations apply to show that $\lim_{n\to\infty} 2^{-l_{\preceq}(x_n, x)} = 2^{-l(x)}$ from the fact that $\lim_{n\to\infty} d_{p_B}(x_n, x) = 0$. Hence $\lim_{n\to\infty} d^s_{q_B}(x, x_n) = 0$. Consequently $(\Sigma^{\infty}, d^s_{q_B})$ is a bicomplete quasi-metric space. Therefore (Σ^{∞}, q_B) is a complete partial quasi-metric space. The proof is complete. ■

17

From now on the pair (Σ^∞, q_B) will be called the Baire partial quasi-metric space.

Notice that

$$q_B(x, y) = 0 \Leftrightarrow l(x) = l(y) = \infty \text{ and } x_k \preceq y_k \text{ for all } k \in \mathbb{N}.$$

So the Baire partial quasi-metric encodes, similarly to the case of the complexity quasi-metric $d_\mathcal{C}$ and the Baire partial metric p_B, the order $\preceq$ on the set Σ and the associated subprefix notion. Furthermore, we wish to emphasize that the Baire partial quasi-metric space remains valid to model all those processes which have been modeled by means of the Baire partial metric space (as, for instance, in program verification or in denotational semantics for programming languages). Nevertheless, the use of he Baire partial quasi-metric presents an advantage with respect to use of the partial metric one, and this advantage is given by its utility, contrarily to the Baire partial metric space, in complexity analysis.

3.2 The asymptotic complexity analysis of algorithms via the Baire partial quasi-metric: Three examples

We end the paper showing that the developed partial quasi-metric theory is useful to analyze the asymptotic complexity of algorithms. Furthermore, we validate our results retrieving as a particular case the well-known asymptotic complexity of Mergesort, Quicksort and Largetwo. To this end, let us recall that, as set out in Section 1, when discussing the running time of computing of Divide and Conquer algorithms usually one has to solve recurrence equations of the form

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ aT(\frac{n}{b}) + h(n) & \text{if } n \in \omega_b \end{cases}, \tag{6}$$

where $a, b \in \mathbb{N}$ with $a, b > 1$, $c \in \mathbb{R}^+$ with $c > 0$ and $h \in \mathcal{RT}$ such that $0 < h(n) < \infty$ for all $n \in \mathbb{N}$.

The asymptotic complexity of those algorithms whose running time of computing is given by the solution of a recurrence equation (6) can be analyzed via the next result which is a Baire partial quasi-metric space version of Theorem 1 in Section 1.

Theorem 8. *Let* $\Sigma = (0, \infty]$ *and* $a, b \in \mathbb{N}$ *with* $a, b > 1$. *Fix* $z \in \Sigma^\infty$ *with* $l(z) = \infty$ *and* $z_k \neq \infty$ *for all* $k \in \omega_b$ *and* $k \geq 2$. *Let* $\Sigma_{b,c}^\infty$ *be the subset of*

Σ^∞ given by $\Sigma_{b,c}^\infty := \{y \in \Sigma^\infty : 2 \leq l(y) \text{ and } y_1 = c, y_k = \infty \text{ for all } k \notin \omega_b \text{ with } 2 \leq k \leq l(y)\}$. Then the mapping $\Theta_{a,b}^z : \Sigma_{*,b}^\infty \to \Sigma_{b,c}^\infty$ defined by $\Theta_{a,b}^z(x) = x_{\Theta_{a,b}^z}$, where

$$
(x_{\Theta_{a,b}^z})_k := \begin{cases} c & \text{if } k = 1 \\ \infty & \text{if } k \notin \omega_b \text{ and } 2 \leq k \leq l(x) + 1 \\ a \cdot x_{\frac{k}{b}} + z_k & \text{if } k \in \omega_b \text{ and } \frac{k}{b} \leq l(x) \end{cases}
$$

has a unique fixed point $v \in \Sigma_{b,c}^\infty$ with $l(v) = \infty$. Moreover if $u \in \Sigma_{b,c}^\infty$ such that $\Theta_{a,b}^z(u) \sqsubseteq_{sp} u$ then $v \sqsubseteq_{sp} u$.

Proof. First of all we note that the subset $\Sigma_{b,c}^\infty$ is closed in $(\Sigma^\infty, d_{q_B}^s)$ and, thus, the pair $(\Sigma_{b,c}^\infty, q_B|_{\Sigma_{b,c}^\infty})$ is a complete partial quasi-metric space. Moreover, the mapping $\Theta_{a,b}^z : \Sigma_{b,c}^\infty \to \Sigma_{b,c}^\infty$ holds the inequality (5) in Theorem 5 with $s = \frac{1}{2}$. So the aforesaid theorem guarantees that $\Theta_{a,b}^z$ has a unique fixed point $v \in \Sigma_{b,c}^\infty$ with $q_B(v, v) = 0$. Hence $l(v) = \infty$.

Now assume the existence of $u \in \Sigma_{b,c}^\infty$ such that $\Theta_{a,b}^z(u) \sqsubseteq_{sp} u$. It follows, by construction of $\Theta_{a,b}^z$, that $l(u) = \infty$. Furthermore, it is clear that $d_{q_B}(\Theta_{a,b}^z(u), u) = 0$. Hence, by Proposition 6, we have that $d_{q_B}(v, u) = 0$. Whence we deduce that $v \sqsubseteq_{sp} u$. $\blacksquare$

Similarly to Section 1, we denote by Φ_T the mapping $\Phi_T : \mathcal{RT}_{b,c} \to \mathcal{RT}_{b,c}$ given by

$$
\Phi_T(f)(n) = \begin{cases} c & \text{if } n = 1 \\ \infty & \text{if } n \notin \omega_b \text{ and } n > 1 \\ af(\frac{n}{b}) + h(n) & \text{otherwise} \end{cases} .
$$

Corollary 9. *A Divide and Conquer recurrence of the form (6) has a unique solution $f_T \in \mathcal{RT}_{b,c}$. Moreover if there exists $g \in \mathcal{RT}_{b,c}$ such that Φ_T is an improver with respect to g, then $f_T \in \mathcal{O}(g)$.*

Proof. Let $v \in \Sigma_{b,c}^\infty$ be the fixed point of the mapping $\Theta_{a,b}^z$ ensured by Theorem 8. Define $f_v \in \mathcal{RT}$ following the same arguments as in Section 2. We immediately obtain that $f_v \in \mathcal{RT}_{b,c}$ is the unique solution f_T to the recurrence equation (6). So f_v can be identified with the running time of computing of a Divide and Conquer algorithm. In addition if Φ_T is an improver with respect to $g \in \mathcal{RT}_{b,c}$, then we can identify such a complexity function with a word $y^g \in \Sigma_{b,c}^\infty$, defined by $y_k^g = g(k)$ for all $k \in \omega_b$, such that

19

$\Theta_{a,b}^z(y^g) \sqsubseteq_{sp} y^g$. It follows, by Theorem 8, that $v \sqsubseteq_{sp} y^g$, that is $f_v \in \mathcal{O}(f_{y^g})$. Since $f_{y^g} = g$ we have obtained that $f_v \in \mathcal{O}(g)$. ∎

Typical examples of algorithms whose running time of computing is the solution to a recurrence equation of kind (6) are Mergesort and (in the best case behaviour).

In the case of Mergesort the recurrence equation (6) in the worst case behaviour is the following:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\frac{n}{2}) + n - 1 & \text{if } n \in \omega_2 \end{cases} , \tag{7}$$

and in the best and the average case behaviour is exactly the next one:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\frac{n}{2}) + \frac{n}{2} & \text{if } n \in \omega_2 \end{cases} . \tag{8}$$

When Quicksort is considered the recurrence equation (6) associated to the running time of computing in the best case behaviour is exactly the following one:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\frac{n}{2}) + dn & \text{if } n \in \omega_2 \end{cases} , \tag{9}$$

where $d \in \mathbb{R}^+$ wiht $d > 0$.

As an immediate consequence of Theorem 8 and Corollary 9 we obtain the following well-known results which ratify, in part, the proposed theory.

Corollary 10. *Let* $r \in \mathbb{R}^+$ *with* $r > 0$. *Define the mapping* $g_{log_2}^r \in \mathcal{RT}_{b,c}$ *by*

$$g_{log_2}^r(n) = \begin{cases} c & n = 1 \\ \infty & n \notin \omega_2 \text{ and } n > 1 \\ rn \log_2 n & \text{otherwise} \end{cases} .$$

Then the running time of computing of

1) Mergesort in the worst case behaviour is in the complexity class $\mathcal{O}(g_{log_2}^1)$.

2) Mergesort in the best and the average case bahviour is in the complexity class $\mathcal{O}(g_{log_2}^{\frac{1}{2}})$.

3) Quicksort in the best case bahaviour is in the complexity class $\mathcal{O}(g_{log_2}^d)$.

Next we show that the techniques based on the Baire partial quasi-metric space are applicable to a more general class of algorithms than those whose running time of computing can be associated with a solution to the recurrence equation (6).

In spite of seeming natural that the complexity analysis of Divide and Conquer algorithms always leads to recurrence equations of type (6), sometimes these kind of recursive algorithms yield recurrence equations that differ from (6). A well-known example of this type of situation is provided by Quicksort. In the worst case behaviour the recurrence equation obtained for Quicksort is given exactly as follows:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ T(n-1) + jn & \text{if } n \geq 2 \end{cases}, \tag{10}$$

where $j \in \mathbb{R}^+$ with $j > 0$. Observe that in this case it is not necessary to restrict the input size of the data to the set ω_b for some $b \in \mathbb{N}$ with $b > 1$.

Another example of algorithms, in this case a non recursive algorithm, whose complexity analysis leads to a recurrence equation different from (6) is the well-known Largetwo. This finds the two largest entries in one-dimensional array of size $n \in \mathbb{N}$ with $n > 1$ (for a deeper discussion see [4]). The running time of computing of Largetwo in the average case behaviour can be associated with the solution to the recurrence equation given as follows:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ T(n-1) + 2 - \frac{1}{n} & \text{if } n \geq 2 \end{cases}, \tag{11}$$

where c is, again, the time taken by the algorithm in the base case, i.e. when the input data is a one-diemensional array with only one element or the array does not contain input data. Notice that Largetwo needs inputs data with size at least 2.

Of course the recurrence equations that yield the running time of computing of the above aforesaid algorithms can be considered as particular cases of the following general one:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ T(n-1) + h(n) & \text{if } n \geq 2 \end{cases}, \tag{12}$$

where $c \in \mathbb{R}^+$ with $c > 0$ and $h \in \mathcal{RT}$ such that $0 < h(n) < \infty$ for all $n \in \mathbb{N}$.

21

Setting
$$\mathcal{RT}_c = \{f \in \mathcal{RT} : f(1) = c\}$$
and defining $\Gamma_T : \mathcal{RT}_c \to \mathcal{RT}_c$ by

$$\Gamma_T(f)(n) = \begin{cases} c & \text{if } n = 1 \\ f(n-1) + h(n) & \text{if } n \geq 2 \end{cases}, \tag{13}$$

we have that similar considerations to those given in the proof of Theorem 8 apply to next one, which gives a method based on $\mathcal{RT}_c$ (Corollary 12) to describe the complexity of those algorithms whose running time of computing satisfies the recurrence equation (12).

Theorem 11. *Let $\Sigma = (0, \infty]$. Fix $z \in \Sigma^\infty$ with $l(z) = \infty$ and $z_k \neq \infty$ for all $k \in \mathbb{N}$ and $k \geq 2$. Let Σ_c^∞ be the subset of Σ^∞ given by $\Sigma_c^\infty := \{y \in \Sigma^\infty : 2 \leq l(y) \text{ and } y_1 = c\}$. Then the mapping $\Psi^z : \Sigma_c^\infty \to \Sigma_c^\infty$ defined by $\Psi^z(x) = x_{\Psi^z}$, where*

$$(x_{\Psi^z})_k := \begin{cases} c & \text{if } k = 1 \\ x_{k-1} + z_k & \text{if } 2 \leq k \leq l(x) \end{cases},$$

has a unique fixed point $v \in \Sigma_c^\infty$ with $l(v) = \infty$. Moreover if $u \in \Sigma_c^\infty$ such that $\Psi^z(u) \sqsubseteq_{sp} u$ then $v \sqsubseteq_{sp} u$.

Corollary 12. *A recurrence of the form (12) has a unique solution $f_T \in \mathcal{RT}_c$. Moreover if there exists $g \in \mathcal{RT}_c$ such that Γ_T is an improver with respect to g, then $f_T \in \mathcal{O}(g)$.*

From Theorem 11 and Corollary 12 we immediately deduce the next well-known results.

Corollary 13. *Let $d, r \in \mathbb{R}^+$ with $d, r > 0$. Then the following assertions hold:*

1) *The running time of computing of Quicksort in the worst case behaviour is in the the complexity class $\mathcal{O}(g_k)$, where $k = \max\left(\frac{c}{4} + \frac{i}{2}, \frac{3j}{5}\right)$ and*

$$g_r(n) = \begin{cases} c & \text{if } n = 1 \\ rn^2 & \text{if } n \geq 2 \end{cases}.$$

2) *The running time of computing of the Largetwo in the average case behaviour is in the the complexity class $\mathcal{O}(g_k)$, where $k = \max\left(\frac{2c+3}{2+2d}, 1\right)$ and*

$$g_r(n) = \begin{cases} c & \text{if } n = 1 \\ r(2(n-1) - \log_2 n + d) & \text{if } n \geq 2 \end{cases}$$

4 Conclusions

Partial metric spaces play a distinguished role in Computer Science. Motivated by this fact we have discussed their usefulness for analyzing the complexity of algorithms. In particular we have shown that the concept of partial quasi-metric space, directly related to the partial metric one, is appropriate to carry out, without convergence assumptions, the asymptotic complexity analysis of algorithms in the spirit of Schellekens. In particular we have constructed the Baire partial quasi-metric from a new prefix notion between words over an alphabet, and we have applied the new partial metric structure, via fixed point arguments, to discuss the asymptotic complexity of algorithms whose running time of computing is typically given by a recurrence equation. The running time of computing of Mergesort, Quicksort and Largetwo has been analyzed as specific examples in order to validate the developed theory.

5 Acknowledgements

References

[1] G. Brassard, P. Bratley, *Algorithms: Theory and Practice*, Prentice Hall, New Jersey (1988).

[2] C.S. Calude, S. Marcus, L. Staiger, *A topological characterization of random sequences,* Inform. Process. Lett. **88**, 245-250 (2003).

[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms,* MIT Press, New York (1990).

[4] P. Cull, M. Flahive, R. Robson, *Difference equations: From rabbits to chaos,* Springer, New York (2005).

[5] B.A. Davey, H.A. Priestley, *Introduction to Lattices and Order,* Cambridge University Press, Cambridge (1990).

[6] G. Kahn, *The semantics of a simple language for parallel processing.* In: Proc. of the IFIP Congress Stockholm, pp. 471-475. Elsevier and North-Holland, Amsterdam (1974).

[7] H.P.A. Künzi, H. Pajooshesh, M.P. Schellekens, *Partial quasi-metrics,* Theoret. Comput. Sci. **365**, 237-246 (2006).

[8] H.P.A. Künzi, *Nonsymmetric distances and their associated topologies: About the origins of basic ideas in the area of asymmetric topology.* In: C.E. Aull and R. Lowen (eds.) Handbook of the History of General Topology vol. 3, pp. 853-968. Kluwer, Dordrecht (2001).

[9] S.G. Matthews, *Partial metric topology,* Ann. New York Acad. Sci. **728**, 183-197 (1994).

[10] S.G. Matthews, *An extensional treatment of lazy data flow deadlock,* Theoret. Comput. Sci. **151**, 195-205 (1995).

[11] S. Oltra, S. Romaguera, E.A. Sánchez-Pérez, *Bicompleting weightable quasi-metric spaces and partial metric spaces,* Rend. Circolo Mat. Palermo **51**, 151-162 (2002).

[12] D. Perrin, J.E. Pin, *Infinite words: Automata, Semigroups, Logic and Games,* Pure and Appl. Math. Series, vol. 141, Elsevier Acad. Press, Amsterdam (2004).

[13] S. Romaguera, M.P. Schellekens, *Quasi-metric properties of complexity spaces,* Topology Appl. **98**, 311-322 (1999).

[14] M. Schellekens, *The Smyth completion: a common foundation for denonational semantics and complexity analysis,* Electronic Notes in Theoret. Comput. Sci. **1**, 211-232 (1995).

[15] D. S. Scott, *Outline of a mathematical theory of computation.* In: Proc. 4th Annual Princeton Conference on Information Sciences and Systems, pp. 169-176 (1970).

[16] D. Scott, *Lattice theory, data types and semantics.* In: Proc. Courant Computer Science Symposium on Formal Semantic of Programming Languages, pp. 66-106, Prentice-Hall, Englewood Cliffs (1972).